(Android) Run Video Chat within your Unity application 📱🎮

Something that makes all games more fun is having the ability to see and speak with other players in the game. Games have been including voice chat services for years but when it comes to video there are limited options especially when using Unity. The best option I've found is the Agora.io Video SDK for Unity available through the Unity Asset Store.

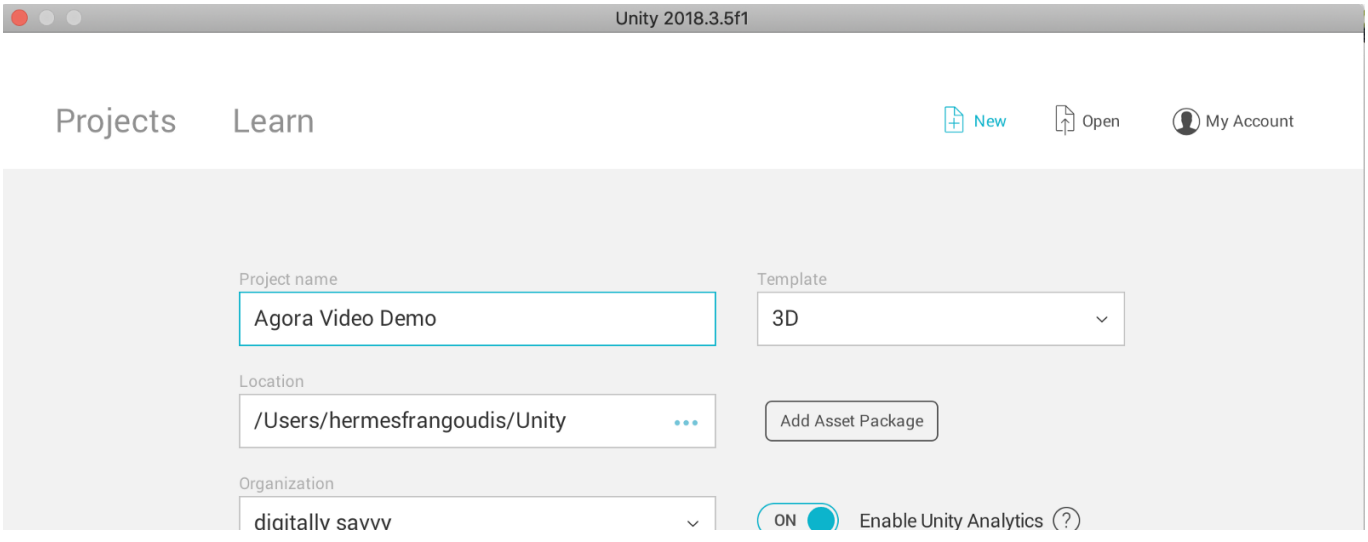
Getting a sample Android app up and running is simple and took less than 10 minutes. Let's run through the steps together.

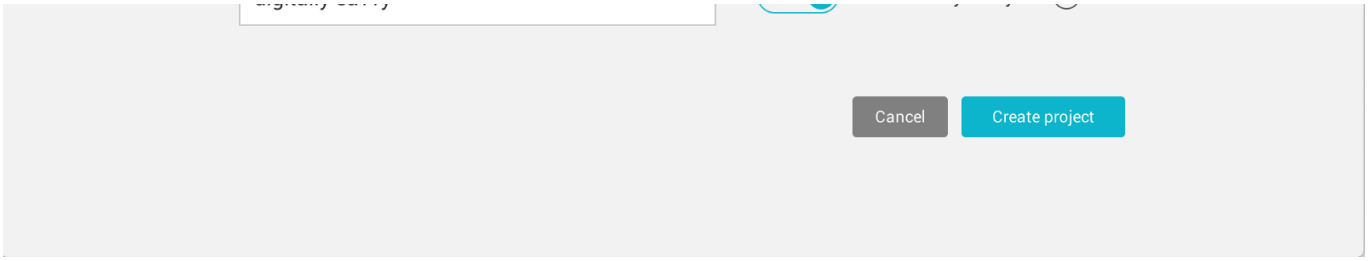
Prerequisites

- Unity Editor
- A developer account with Agora.io

Getting Started

To start, open Unity and create a blank new project named **Agora Video Demo**.

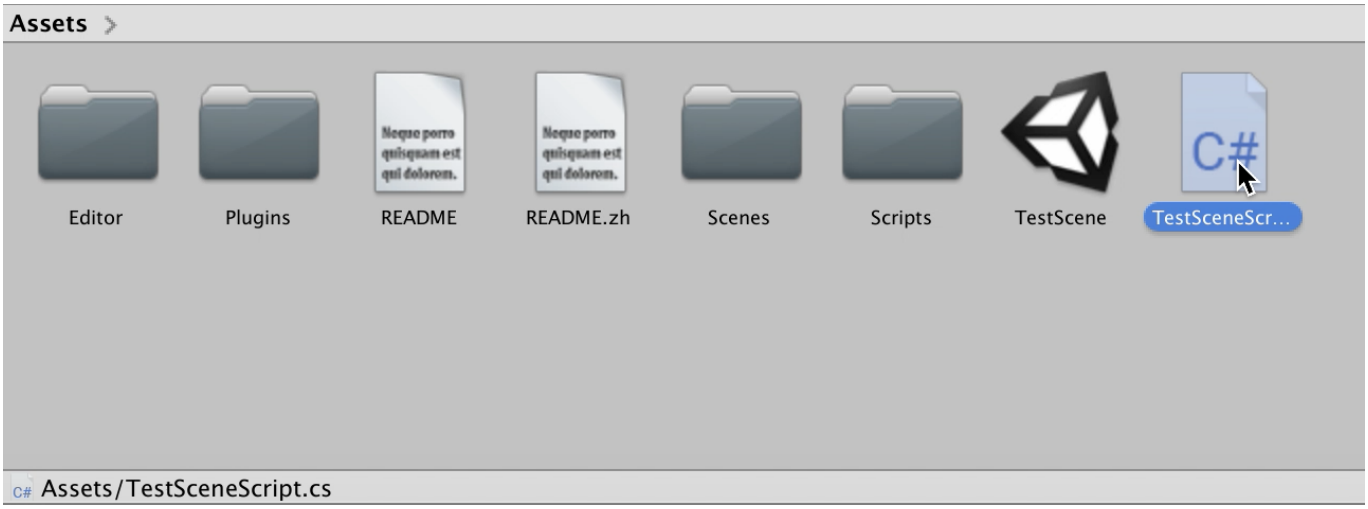




The next step is to navigate to the Unity Store (*if you are in the scene view, click the Unity Store tab*) and search for "Agora voice SDK". Once the plugin page has loaded, go ahead and click ***Download***. Once the plugin page has loaded, go ahead and click ***Download***. Once the download is complete, click and ***Import*** the assets into your project.

Update code

Great! Now that we have the plugin downloaded and imported to our project we need to make 1 quick update before we can build and run our app. Double click the `TestSceneScript.cs` file within the assets panel to edit it.



The file will open in Visual Studio (*Unity's default code editor*). Let's scroll down to `line 19`.

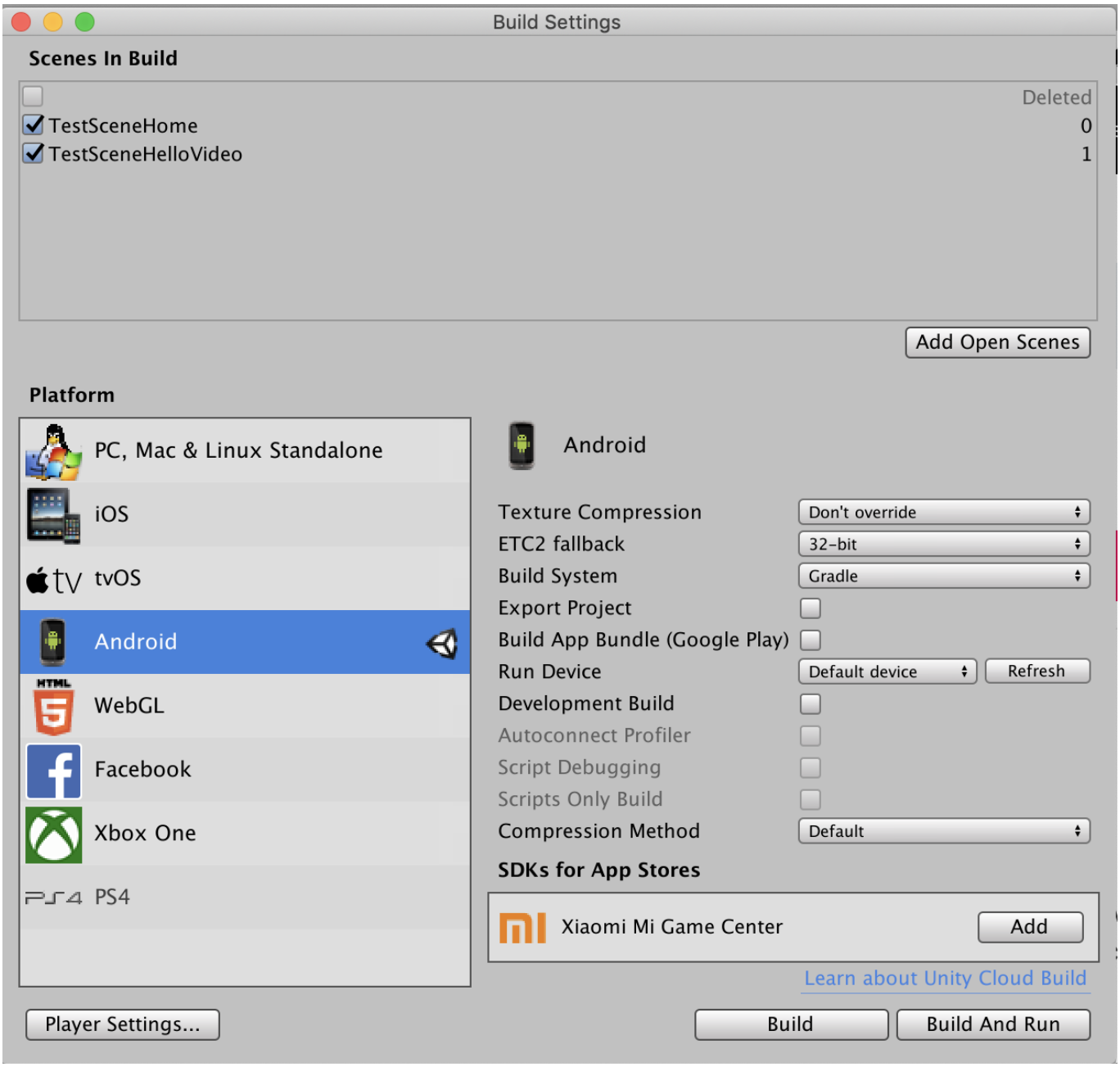
```
  private string appId = "YOUR APP ID";
```

At this point we need to log into our Agora.io developer account and create a new AppID or select an existing one. I had previously created one so I'll copy paste this appid into the `" "` on `line 19`. Make sure to save your changes.

We can scroll through the `TestSceneScript.cs` file to see all the various event callbacks the Agora.io Voice SDK for Unity offers.

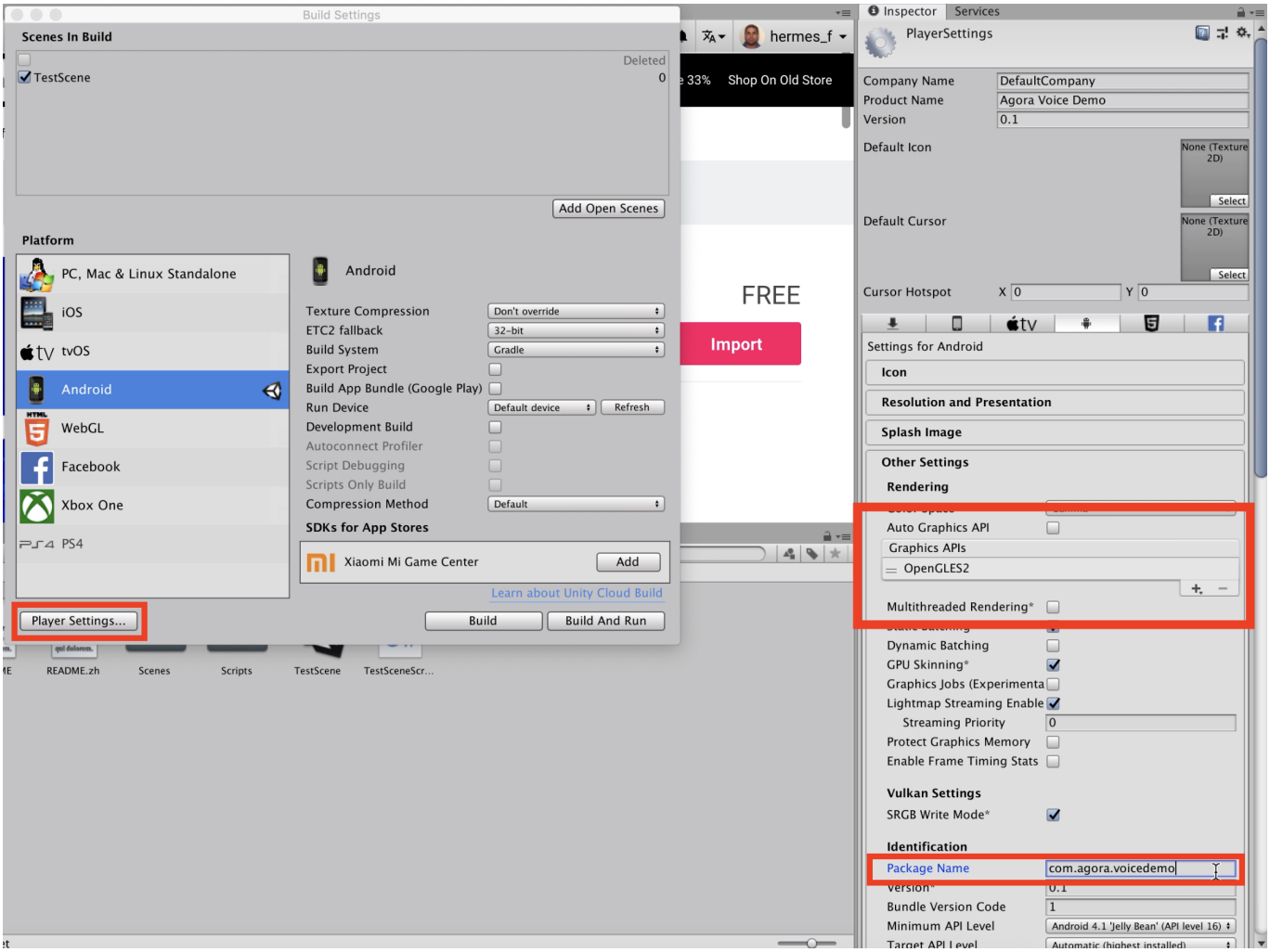
Player Settings for Building the Sample Application

Open the **Build Settings** and drag the `TestSceneHome.unity` & `TestSceneHomeWorld.unity` scenes from the assets list into the "*Scenes in Build*" list. Select ***Android*** from the *Platform* list and click ***Switch Platform***.



Once Unity finishes its setup process, open the ***Player Settings***. We need to make some changes to the player settings. First deselect *Multithreaded Rendering* option and deselect

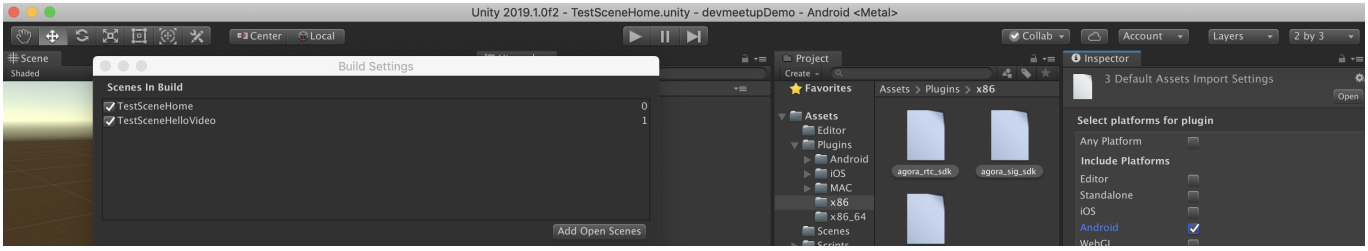
the *Auto Graphics API*. Next, within the *Graphics API* options that are now visible, make sure only OpenGLES2 is on the list (*you may need to add it using the `+` icon*) and remove any other APIs that may appear on the list.

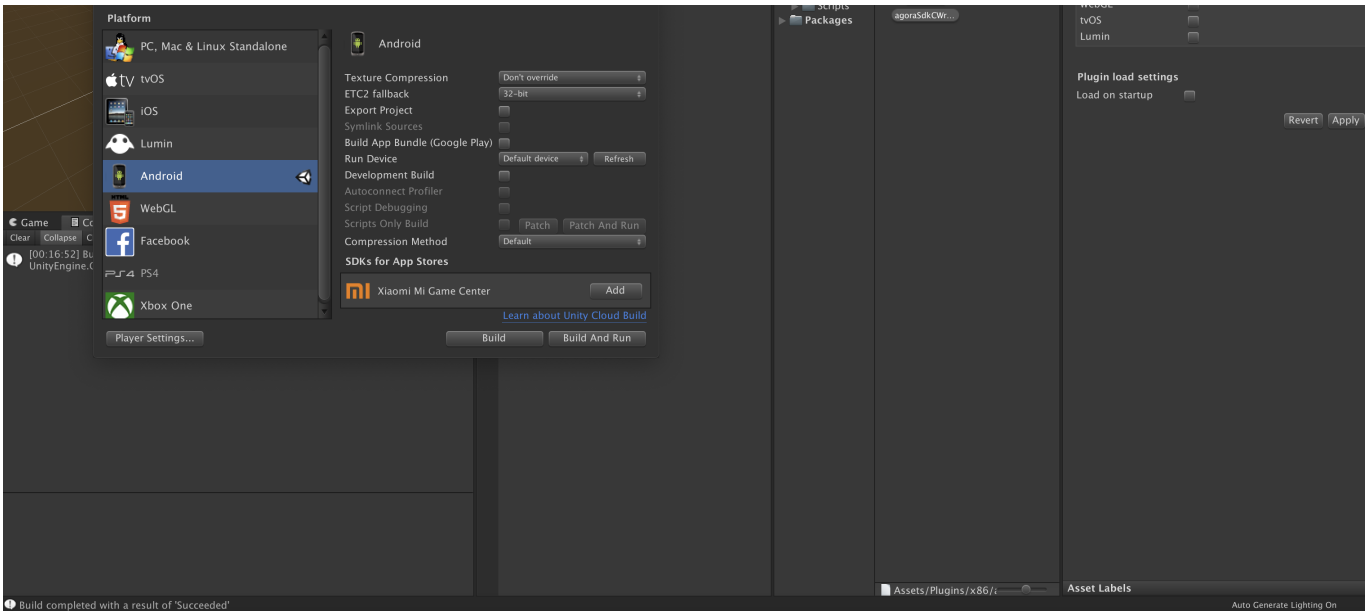


Be sure our Android app also has a unique package name, I chose `com.agora.voicedemo`.

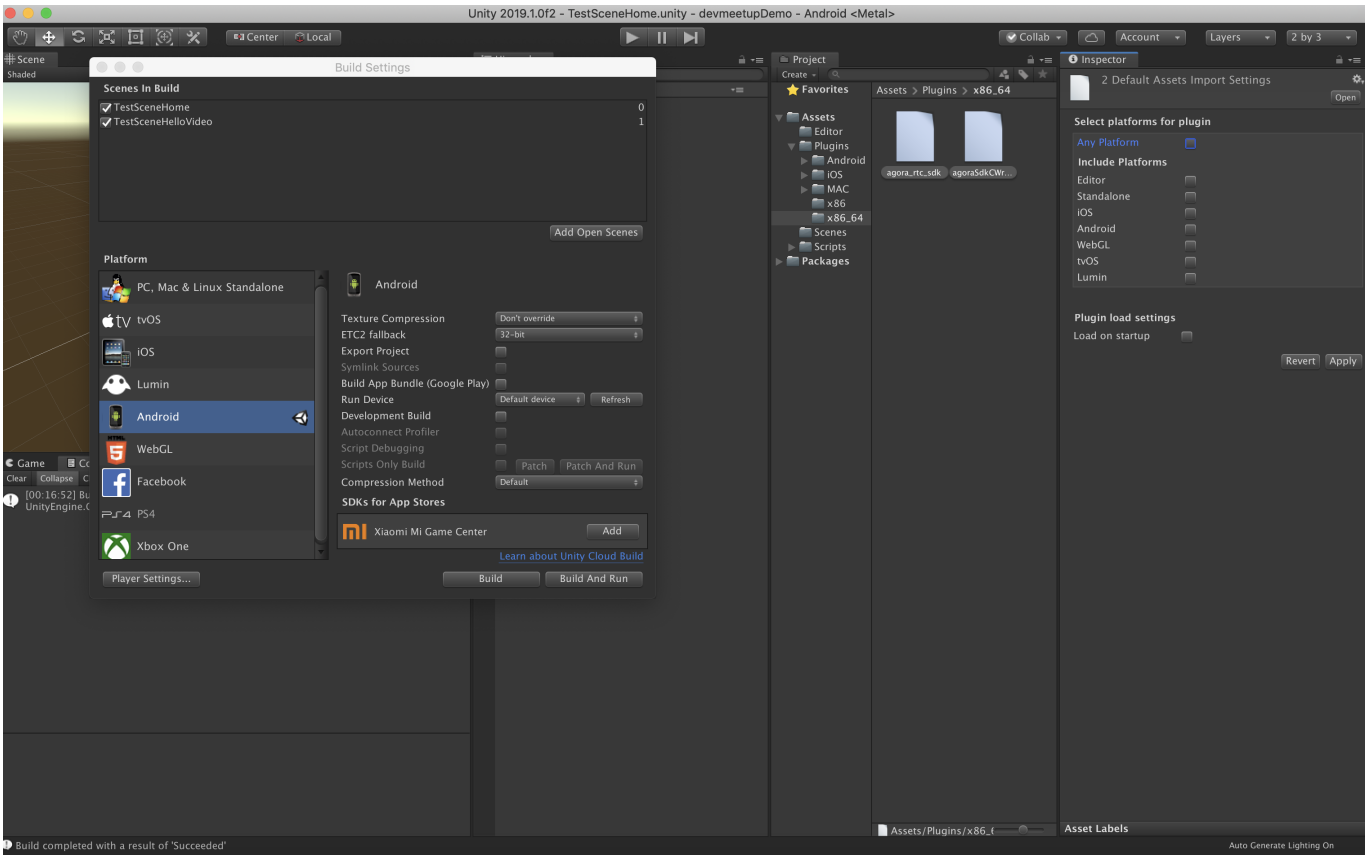
Set Plugin Identities

The Agora.io Video Unity SDK has cross-functionality with Android, iOS, Mac, and Windows we need to set the plugin identities. Along with identities we will also need to also choose between **x86** and **x86_64** identities.





Let's start by going into *Assets>Plugins>x86* folder, select only **Android** and then click *Apply*.



To ensure that the various build plugins don't collide we will then need to go into *Assets>Plugins>x86_64,* select all the files. Next go into the Inspector, uncheck all items and click *Apply*.

Make sure your Android device is plugged and click ***Build And Run***. Unity will prompt you to save your project, I named my build *Android*.

All Done!

Thanks for following along, feel free to leave a comment below!

Other Resources

- The complete API documentation is available in the Document Center.
- For technical support, submit a ticket using the Agora Dashboard or reach out directly to our Developer Relations team devrel@agora.io

> *Note: Older versions of the Agora Unity SDK shipped with gradle that can cause compile errors. If you run into an issue when building for Android, where Unity throws an error "*Unable to build gradle,*" open your file explorer and navigate into the project* Assets > Plugins > Android *and delete the* `mainTemplate.gradle` *file.*